Volume 13. No 2, May 2026
e-ISSN 2527-9572 / p-ISSN 2354-8924

A Web-Based Violence Reporting Application for Women and Children: A Case Study of the SP Sambe Women's School Community in Tiley Pantai Village Using the Waterfall Development Model

Imam Hizbullah
Department of Electrical Engineering, Khairun University, Ternate, Indonesia.
Jl. Jusuf Abdulrahman, Gambesi, Kec. Ternate Selatan
Kota Ternate, Maluku Utara
imam@unkhair.ac.id

Muhammad Natsir Rahman
Department of Electrical Engineering, Khairun University, Ternate, Indonesia.
Jl. Jusuf Abdulrahman, Gambesi, Kec. Ternate Selatan
Kota Ternate, Maluku Utara
Mnr4hm4n@gmail.com

Hafid Syaifudin
Department of Electrical Engineering, Khairun University, Ternate, Indonesia.
Jl. Jusuf Abdulrahman, Gambesi, Kec. Ternate Selatan
Kota Ternate, Maluku Utara
Hafiz.zaifudin@gmail.com

Article history: Received May 7, 2026| Revised May 12, 2026| Accepted May 17, 2026

Abstract **– Violence against women and children constitutes a serious and escalating social problem; however, reporting rates remain critically low due to the absence of safe, accessible, and confidential reporting mechanisms. This study aimed to develop a web-based violence reporting application for the SP Sambe Women's School (Sekolah Perempuan/SP) community in Tiley Pantai Village, Morotai Island Regency. The application was developed using the waterfall software development model, which encompasses four sequential phases: requirements analysis, system design, development, and testing. Data were collected through structured observation and in-depth interviews with prospective end-users. The results indicate that the application's core features include anonymous incident reporting, report tracking via unique system-generated codes, role-specific dashboards for administrators and officers, and a case follow-up management module. Black-box testing confirmed that all implemented features functioned in accordance with the specified requirements. The application is expected to facilitate rapid, secure, and confidential reporting for community members, while simultaneously improving the efficiency of case management and administrative oversight for the SP Sambe community. This study affirms that the strategic utilisation of information technology can meaningfully contribute to the resolution of complex social problems, particularly those pertaining to violence against women and children.**

Keywords: information system; reporting application; violence against women and children; waterfall model; web-based system

I. INTRODUCTION

Sexual violence represents one of the most severe violations of human rights, inflicting profound physical, psychological, and social consequences upon its victims. In Indonesia, data published by the National Commission on Violence Against Women (Komisi Nasional Anti Kekerasan terhadap Perempuan/Komnas Perempuan) indicate that the incidence of sexual violence has increased consistently on an annual basis [1]. This trend is corroborated at the local level: the Morotai District Attorney's Office reported that cases of violence and sexual harassment against women and children constituted the largest category of cases handled in 2025, with the majority of victims being minors. Despite the prevalence of such incidents, a substantial proportion of cases remain unreported. Contributing factors include victims' fear of retaliation, pervasive social stigma, distrust of formal legal processes, and the critical absence of reporting channels that are simultaneously safe, confidential, and easily accessible.

The rapid advancement of information and communication technology presents considerable opportunities for addressing a broad range of social challenges [2] . In the context of violence against women and children, technology-mediated solutions[3] specifically web-based reporting applications offer a viable mechanism for enabling victims to document and report incidents in a swift, secure, and anonymous manner.

Several prior studies have examined the development of technology-based reporting systems. Rahayu et al. [4] developed an Android-based complaint application for violence against women and

children in Sukabumi, employing use case diagrams, activity diagrams, and class diagrams. The resulting prototype enabled mobile users to submit reports and visualise the type and location of violent incidents. In a related study, Fajria [5]developed an Android-based application for the Ministry of Women's Empowerment and Child Protection using the Systems Development Life Cycle (SDLC) approach, producing an integrated platform capable of generating real-time reports across multiple service units. Cahyani et al.[6-7] designed an Android-based platform for reporting sexual violence within a university campus environment, which demonstrated a positive effect on students' willingness to report incidents and facilitated institutional prevention and response mechanisms. Furthermore, a study conducted in Aceh [8] analysed and designed an Android-based reporting application for violence against women and children using PHP and the CodeIgniter framework, significantly streamlining the reporting process within that region.

Collectively, the aforementioned studies demonstrate the substantial potential of information technology in addressing violence-related social problems. Building upon this body of literature, the present study developed a web-based reporting application specifically tailored to the operational requirements of the SP Sambe Women's School community in Tiley Pantai Village, Morotai Island Regency. This application is intended to serve as a contextually appropriate technological solution for facilitating the reporting of violence against women and children within this community.

This study adopted the waterfall software development model as its methodological framework. The waterfall model is a sequential, plan-driven approach in which each development phase must be completed before the subsequent phase commences [9]. Data were collected through structured observation and semi-structured interviews with prospective end-users prior to system development. The four phases of the waterfall model applied in this study are described as follows:

Requirements Analysis. This phase focused on the comprehensive elicitation and definition of user and system requirements prior to the commencement of development. Activities included semi-structured interviews, direct observation, document review, and business process analysis, all aimed at establishing a thorough understanding of the functional and non-functional expectations of the software.

System Design. In this phase, the requirements established in the preceding stage were translated into detailed technical specifications governing system construction. The design artefacts encompassed system architecture, database schema, and Unified Modelling Language (UML) diagrams [10]. These documents served as authoritative references for the development team, ensuring consistency and alignment with user requirements throughout the implementation process.

Development. This phase involved the transformation of the design specifications into functional, executable program code in accordance with the technical documentation produced in the design phase.

Testing. System testing was conducted to verify that the developed application operated correctly and in full accordance with the specified requirements. Testing was carried out using the black-box testing approach, which evaluates system behaviour against defined functional requirements without reference to the internal code structure.

It is noted that the scope of the present study encompasses all four phases of the waterfall model, from initial requirements analysis through to system testing.

II. METHOD

This study adopted the waterfall software development model as its methodological framework. The waterfall model is a sequential, plan-driven approach in which each development phase must be completed before the subsequent phase commences [9], [11], [12]. Data were collected through structured observation and semi-structured interviews with prospective end-users prior to system development. The four phases of the waterfall model applied in this study are described as follows:

Requirements Analysis. This phase focused on the comprehensive elicitation and definition of user and system requirements prior to the commencement of development. Activities included semi-structured interviews, direct observation, document review, and business process analysis, all aimed at establishing a thorough understanding of the functional and non-functional expectations of the software.

System Design. In this phase, the requirements established in the preceding stage were translated into detailed technical specifications governing system construction. The design artefacts encompassed system architecture, database schema, and Unified Modelling Language (UML) diagrams. These documents served as authoritative references for the development team, ensuring consistency and alignment with user requirements throughout the implementation process.

Development. This phase involved the transformation of the design specifications into functional, executable program code in accordance with the technical documentation produced in the design phase.

Testing. System testing was conducted to verify that the developed application operated correctly and in full accordance with the specified requirements. Testing was carried out using the black-box testing

approach, which evaluates system behavior against defined functional requirements without reference to the internal code structure.

It is noted that the scope of the present study encompasses all four phases of the waterfall model, from initial requirements analysis through to system testing.

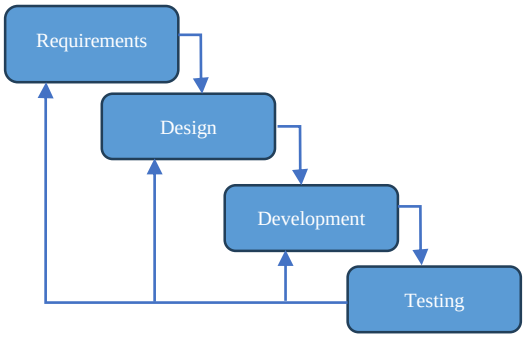

Figure 1. Phase Waterfall Model.

III. RESULTS AND DISCUSSION

A. *Requirements Analysis*

The requirements analysis phase, conducted through structured observation and user interviews, identified two categories of system requirements: functional requirements and non-functional requirements.

1. Functional Requirements

Functional requirements define the specific services and features that the system must provide for each user role (actor).

For the General Public / Reporters:

- Self-Reporting: The system shall provide an incident reporting form accessible without user authentication.
- Anonymity: The system shall permit report submission without requiring the reporter to disclose their identity, thereby supporting the anonymous reporting option.
- Automatic Tracking Code Generation: Upon successful report submission, the system shall automatically generate and display a unique alphanumeric code (e.g., SPB-2026-0001).
- Report Tracking: The system shall provide a search function that allows users to retrieve the current status and handling history of a previously submitted report using the unique tracking code.
- Educational Information: The system shall display guidance on the reporting procedure, relevant emergency contact numbers, and a Frequently Asked Questions (FAQ) section.

For Officers (Village-Level Response Officers):

- Session Management: Officers shall be able to authenticate and terminate their sessions securely.
- Personal Dashboard: Officers shall have access to a summary view displaying the number of reports currently assigned to them.
- Report Details: Officers shall be able to view the complete details of incidents, victims, and reporters for cases assigned to their care.
- Follow-Up Documentation: Officers shall be able to record chronological notes detailing the case management steps undertaken (e.g., home visits, medical assistance facilitation).
- Status Update: Officers shall be able to update a report's status from 'In Progress' to 'Resolved'.

For Administrators (Community Administrators):

- Comprehensive Report Access: Administrators shall have unrestricted visibility of all reports submitted to the system.
- Officer Dispatching: Administrators shall be able to assign specific officers to individual reports.
- User Account Management: Administrators shall be able to create, deactivate, and reset the passwords of officer accounts.
- Data Visualization: Administrators shall have access to graphical representations (Donut Charts) of statistical distributions of violence categories for internal reporting purposes.
- Report Moderation: Administrators shall possess the authority to reject reports deemed invalid or fraudulent.

2. Non-Functional Requirements

Non-functional requirements define the system's quality attributes, operational constraints, and technical standards.

Usability:

- Rural Accessibility: The user interface shall employ straightforward Bahasa Indonesia, avoiding technical jargon to accommodate users with limited digital literacy.
- Typography and Visual Contrast: The system shall utilize a large typeface (Nunito, 17px) and high-contrast color schemes to ensure readability for elderly users and those operating in high-ambient-light environments.
- Mobile-First Responsive Design: Given the predominant use of smartphones among the target population, the interface shall be fully responsive and optimised for touch-based navigation.
- Psychologically Informed Colour Scheme: A warm and reassuring colour palette (Teal and Amber) shall be adopted to convey a sense of safety and approachability for victims.

Security:

- Password Encryption: All user passwords shall be stored using a one-way cryptographic hashing algorithm (Bcrypt).
- Input Sanitisation: The system shall implement protections against SQL Injection attacks (via PDO Prepared Statements) and Cross-Site Scripting (XSS) (via HTML entity encoding).
- Role-Based Access Control: Access to administrative and officer dashboards shall be

enforced through rigorous server-side session verification

- Data Confidentiality: Victim's personal data shall not be exposed on publicly accessible pages; such data shall be retrievable only through the unique tracking code or by authorised personnel.

B. Design

The system design phase employed standard modelling tools to translate the elicited requirements into technical specifications. A Use Case Diagram was constructed to represent the interactions between user roles (actors) and the system's functional components. An Entity Relationship Diagram (ERD) was subsequently developed to model the relational structure of the system's database, illustrating the associations among the core data entities. These artefacts provided a structured and unambiguous specification for the subsequent development phase.

1. Use Case Diagram

The violence reporting system's use case diagram can be seen in. figure 2.

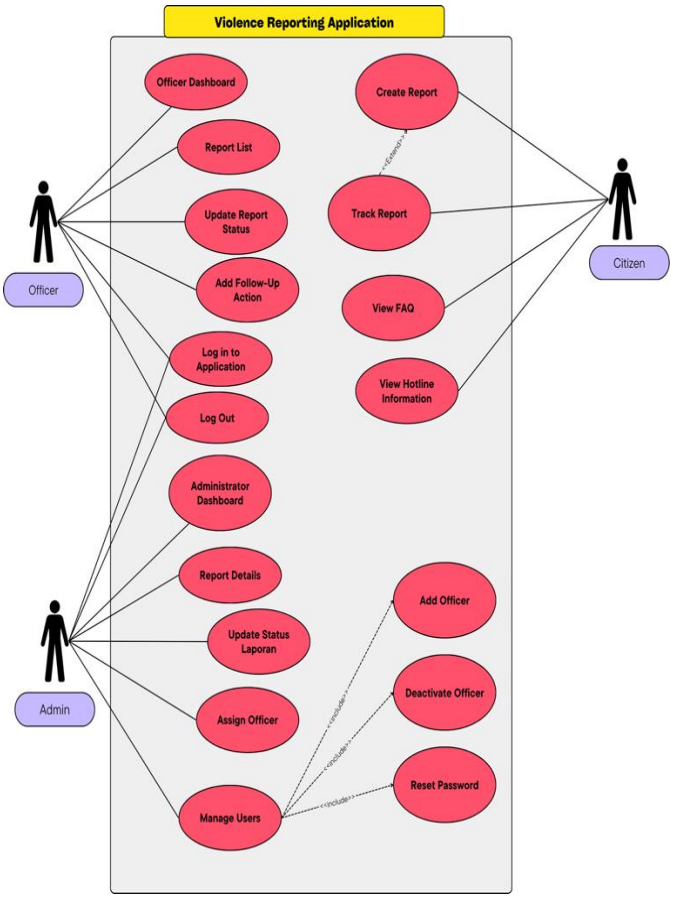

Figure 2. Phase Waterfall Model.

2. Entity Relationship Diagram

The database design is presented in an Entity Relationship Diagram (ERD) to illustrate the associations among tables within the developed system's database. The Entity Relationship Diagram of the violence reporting system is presented in Figure 3

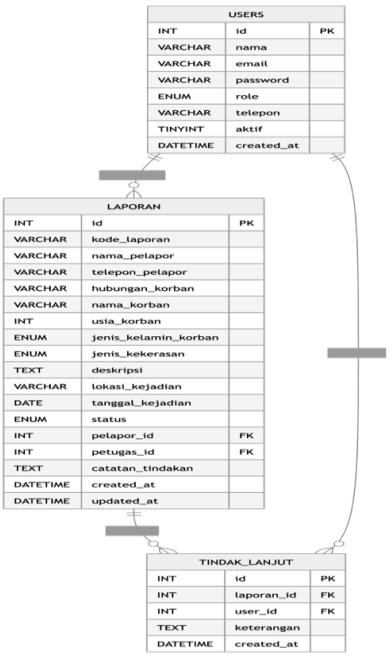

Figure 3. Entity Relationship Diagram.

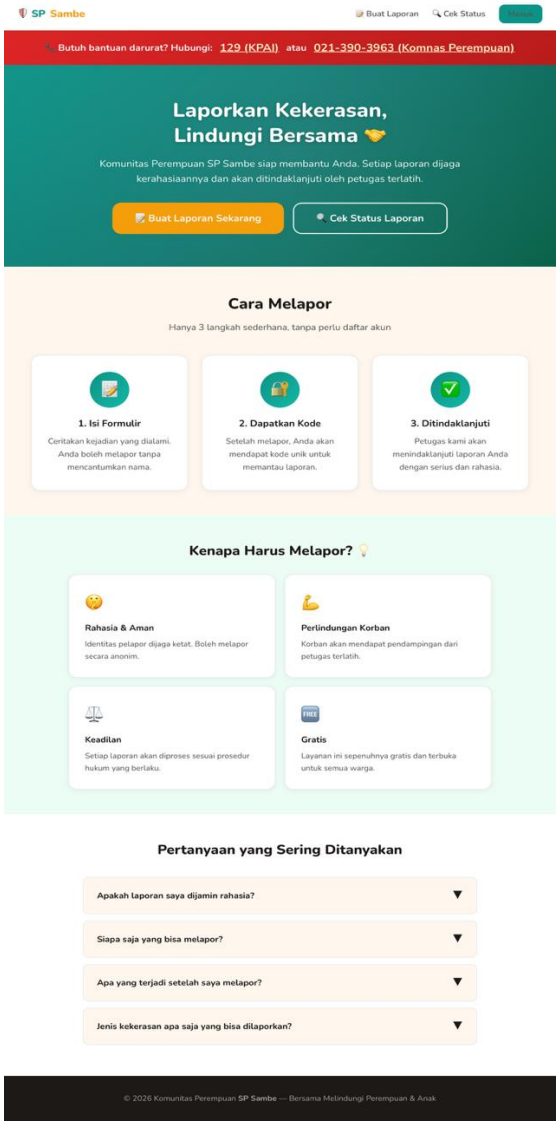

Figure 4. Landing Page.

C. *Development*

The development of the reporting system was guided by the features identified during requirements analysis and the design created in the previous stage. Figure 4. shows the landing page displayed when the application is accessed by residents, officers, and administrators. From this page, residents can access brief information on how to file a report and important hotlines for emergencies. Residents can also submit reports without logging in and track previously submitted reports.

The login page, presented in Figure 5, serves as the authentication page through which officers and administrators access their respective dashboards. The dashboard pages for officers and administrators are presented in Figures 6 and 7 respectively.

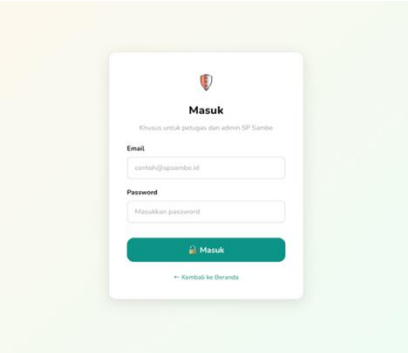

Figure 5. Login Page

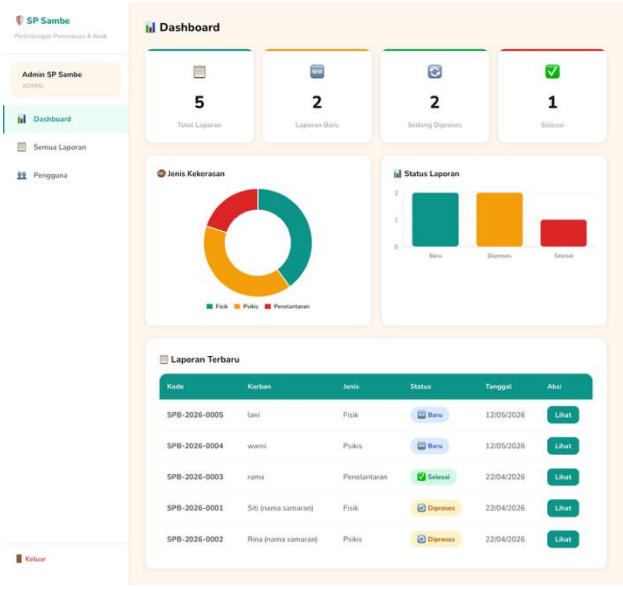

Figure 6. Admin Dashboard

Administrator Dashboard features report management and the ability to assign officers to handle reports. The admin can also manage users — registering new admins and new officers.

The Officer Dashboard, while sharing structural similarities with the Administrator Dashboard, is differentiated by the scope of its accessible features: officers may update report statuses, document case follow-up actions, and review the details of reports assigned to them, but do not have access to user management or system-wide statistical visualisations.

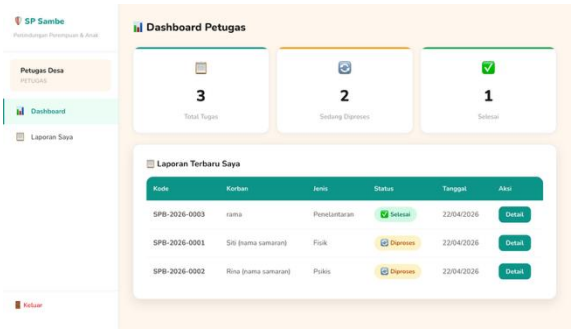

Figure 7. Officers Dashboard

D. *Testing*

System testing was conducted using the black-box testing approach, which evaluates the conformance of system outputs to predefined functional requirements based solely on specified inputs, without examining the underlying source code [8]. The testing encompassed four modules: the public-facing module, the authentication module, the administrator feature module, and the officer feature module. The results of each testing module are presented in the tables below.

Table 1. Black-Box Testing Results: public Modul

Feature	Test Scenario	Input Data	Expected Outcome
Submit Report (Complete)	Submit a report with all required fields correctly completed.	Name, victim details, type, description, etc.	The system saves the data, displays a success message, and generates a Report Code.
Submit Report (Anonymous)	Submit a report without providing reporter identity.	Reporter name/phone left blan k.	The system successfully saves the report as 'Anonymous' and generates a code.
Form Validation	Submit a report with required fields (*) left empty.	Description empty / type not selected.	An error message is displayed: 'Please fill in the victim's name, type of violence, and description.'
Tracking (Found)	Search for report status using a valid code.	Report code (e.g., SPB-2026-0001).	The system displays report details, current status, and follow-up history.
Tracking (Not Found)	Search for report status using an invalid or random code.	'WRONG-CODE-123'.	An error message is displayed: 'No report found with the given code.'

As presented in Table 1, testing of the public module confirmed that all five test scenarios — including complete report submission, anonymous report submission, form validation, and report tracking (both successful and unsuccessful) — produced outcomes consistent with the specified requirements.

Table 2. Black-Box Testing Results: Authentication Modul

Feature	Test Scenario	Input Data	Expected Outcome
Successful Login	Login using correct email and password.	admin@spsambe.id / password123	User is successfully directed to the dashboard corresponding to their role.
Failed Login	Login using an incorrect password.	admin@spsambe.id / wrongpass123	An error message is displayed: 'Incorrect email or password.'
Logout	Log out of the system.	Click the Logout button.	Session is terminated and the user is redirected to the login page.
URL Protection	Directly access /admin/ without prior authentication.	Access URL via browser address bar.	The system denies access and redirects the user to the login page.
Role-Based Access Control	Officer attempts to access /admin/users.php.	Officer accesses an admin-only URL.	An 'Access Denied' message is displayed (HTTP 403).

Table 2 presents the authentication testing results, which confirmed the effective enforcement of role-based access control, including the prevention of unauthorised direct URL access and the restriction of officer access to administrator-only pages.

Table 3. Black-Box Testing Results: Administrator Feature Module

Feature	Test Scenario	Input Data	Expected Outcome
Officer Assignment	Admin assigns an officer to a specific report.	Select officer name in report detail view.	Report status changes to 'In Progress'; the assigned officer is displayed in the report details.
Status Update	Admin updates a report status to 'Resolved'.	Select 'Resolved' status and add notes.	Status is updated in the database and reflected on the public tracking page.
User Management	Admin registers a new officer account.	Name, new email address.	New account is created and available for login.
Password Reset	Admin resets an officer's password.	Click the Reset button.	Officer's password is reverted to the system default.
Statistics	View the donut chart on the admin dashboard.	—	The chart is rendered and data are synchronised with the number of reports per category.

As shown in Table 3, all administrator features — encompassing officer dispatching, status updates, user account management, password reset functionality, and statistical data visualisation — functioned as specified.

Table 4. Black-Box Testing Results: Officer Feature Module

Feature	Test Scenario	Input Data	Expected Outcome
Task List	Officer views the list of assigned reports.	—	Only reports assigned to the respective officer are displayed.
Case Follow-Up	Officer adds a case handling note.	'Conducted a field visit to the victim's residence.'	The note is saved and appears in the report's chronological history (timeline).
Case Resolution	Officer updates the report status to 'Resolved'.	Select 'Resolved' status.	The report is reclassified under 'Resolved' in the officer's dashboard.

Table 4 confirms that the officer-facing features, including the task list, case follow-up documentation, and case resolution functionality, operated correctly in accordance with the defined requirements.

E. Discussion

The findings of this study demonstrate that a web-based violence reporting application tailored to the specific operational context of the SP Sambe community can be successfully designed and implemented using the waterfall development model. The application's primary functionalities anonymous incident reporting and real-time report tracking directly address the barriers to reporting identified in the initial requirements analysis, namely the lack of safe and confidential reporting channels. The administrative features, including the officer dispatching system and case follow-up management module, are anticipated to substantially improve the community's capacity for systematic case governance and internal administrative reporting.

The results of black-box testing confirmed that the application's complete feature set operates in full conformance with the user requirements elicited during the analysis phase. These findings are consistent with those of prior studies [5], [13], [14] which have collectively established the efficacy of information technology in facilitating practical reporting activities. However, the present application is distinguished from comparable systems by the depth of its contextualisation: its features and workflows were designed to reflect the specific organisational structure and community dynamics of the SP Sambe community, rather than to serve as a generic, multi-purpose reporting platform.

The adoption of the waterfall model as the development methodology in this study further corroborates the established view that this approach is particularly well-suited to projects characterised by stable, clearly defined, and well-understood requirements [12], [15]. The structured, sequential nature of the waterfall model facilitated systematic documentation and traceability throughout the development process. More broadly, this study contributes to the growing body of evidence supporting the application of information technology as an instrument of social governance and community empowerment, particularly in rural and underserved regions such as Morotai Island Regency.

Nevertheless, the present study is subject to certain limitations. Most notably, the application does not currently incorporate an automated notification mechanism to alert reporters of updates to the status of their submitted cases. Consequently, reporters are required to actively and periodically access the application to monitor case progress. This limitation reflects the early-stage nature of the current implementation and identifies a clear direction for future development.

IV. CONCLUSION

This study successfully developed a web-based violence reporting application for women and children, tailored to the needs of the SP Sambe Women's School community in Tiley Pantai Village, Morotai Island Regency, using the waterfall software development model. The application fulfils the identified user requirements through a suite of features comprising anonymous incident reporting, unique-code-based report tracking, and role-differentiated case management dashboards for administrators and response officers. Comprehensive black-box testing confirmed that all system functions performed correctly and in full accordance with the specified requirements. The application is therefore expected to facilitate more rapid, secure, and confidential incident reporting for community members, while enhancing the overall effectiveness of case management by the SP Sambe organisation. For future development iterations, several enhancements are recommended. First, the integration of an automated notification system would enable reporters to receive real-time updates on the progress of their submitted cases, eliminating the need for active monitoring. Second, the development of a dedicated mobile application version would further improve system accessibility for a smartphone-dependent user population. Third, the reinforcement of security infrastructure including the implementation of multi-factor authentication and comprehensive data encryption would strengthen user data protection. Fourth, formal integration with relevant governmental and law enforcement agencies would accelerate case escalation and resolution. Finally, extended usability and performance evaluations involving a larger and more representative sample of end-users are recommended to ensure the continued improvement of the system's quality and societal impact.

REFERENCES

[1] Y. Rony and H. Yusuf, "Tinjauan Kriminologis Terhadap Kekerasan Seksual Pada Perempuan," *Jicn*, vol. 1, no. 2, pp. 1868–1877, 2024.

[2] B. Jie, E. Eric, D. Mervyn, V. Anggrianto, and K. Kelvin, "Pemanfaatan Dan Dampak Penggunaan Teknologi Informasi Pada Bidang Sosial," *Journal of Information System and Technology*, vol. 4, no. 2, pp. 392–397, Jul. 2023, doi: 10.37253/joint.v4i2.6298.

[3] U. Fatima, N. Ahmad, A. N. Menegaki, and H. Wang, "How do information and communication technologies (ICTs) empower women?," Mar. 01, 2025, *Elsevier Ltd*. doi: 10.1016/j.wsif.2025.103057.

[4] S. Rahayu, M. Rai, and I. Iman, "Rancang Bangun Aplikasi Pengaduan Kekerasan Anak dan Perempuan Berbasis Android," in *SEMNASTERA*, Oct. 2022.

[5] F. Fajria, "Aplikasi Pencatatan Dan Pelaporan Kekerasan Berbasis Android Pada Kementrian Pemberdayaan Perempuan Dan Perlindungan Anak," *JIKOMSTIK*, vol. 8, no. 2, 2022.

[6] R. D. Cahyani, M. R. A. Maklin, S. Karenina, S. Rabbani, A. Sumawidjaya, and R. Setyadinsa, "Perancangan Aplikasi Mobile 'ProtectU' untuk Pelaporan Kekerasan Seksual di Lingkungan Universitas Berbasis Android Studio," *IFTK*, vol. 20, no. 1, Apr. 2023.

[7] R. D. Cahyani, M. R. A. Maklin, S. Karenina, S. Rabbani, A. Sumawidjaya, and R. Setyadinsa, "Perancangan Aplikasi Mobile 'ProtectU' untuk Pelaporan Kekerasan

Seksual di Lingkungan Universitas Berbasis Android Studio," *JURNAL INFORMATIK*, vol. 20, Apr. 2024.

[8] S. Hadiardri and R. Albar, "Analysis Of The Effectiveness Of Designing An Application For Reporting Cases Of Violence Against Women And Children In The Office Of The Technical Implementation Unit Of Protected Area Aceh Women And Children (Uptd Ppa) Android Based," *Journal of Informatics and Computer Science*, vol. 10, no. 1, Apr. 2024.

[9] A. Saravanos and M. X. Curinga, "Simulating the Software Development Lifecycle: The Waterfall Model," *Applied System Innovation*, vol. 6, no. 6, Dec. 2023, doi: 10.3390/asi6060108.

[10] J. Kniežová and P. Hrkút, "UML-Based Modeling for Effective Implementation of Constraint-Type Business Rules in Software Development," in *Procedia Computer Science*, Elsevier B.V., 2025, pp. 1060–1066. doi: 10.1016/j.procs.2025.03.138.

[11] M. Kramer, "Lifecycle : an analyses based on the waterfall model," *Review of Business & Finance Studies*, vol. 9, no. 1, pp. 77–84, 2018.

[12] G. B. M. T. Sitorus, R. A. G. Efendi, J. F. Napitupulu, H. Pranoto, and E. S. Hermawan, "Bridging the Gap: A Comparative Evaluation of Agile, Waterfall, and Hybrid Methodologies in Modern Software Projects," in *Procedia Computer Science*, Elsevier B.V., 2025, pp. 1259–1268. doi: 10.1016/j.procs.2025.09.067.

[13] P. A. Falah, A. Putra Kharisma, and E. Santoso, "Pengembangan Aplikasi Pelaporan Tindak Kekerasan Seksual Di Lingkungan Universitas Brawijaya Berbasis Website," 2017. [Online]. Available: http://j-ptiik.ub.ac.id

[14] W. Permatasari and I. Swasti, "Pemanfaatan Aplikasi 'Siap Ppak' Sebagai Bentuk Pelaporan Kekerasan Pada Perempuan Dan Anak Di Kota Surabaya," *KARYA Jurnal Pengabdian Masyarakat*, vol. Vol 3, no. No 3, 2023.

[15] A. M. Dima and M. A. Maassen, "From Waterfall to Agile software: Development models in the IT sector," *Journal of International Studies*, vol. 11, no. 2, pp. 315–326, 2018, doi: 10.14254/2071.